PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04J

A2

(11) International Publication Number: WO 97/01895

(43) International Publication Date: 16 January 1997 (16.01.97)

(21) International Application Number: PCT/CA96/00431

(22) International Filing Date: 27 June 1996 (27.06.96)

(30) Priority Data:
9513024.1 27 June 1995 (27.06.95) GB

(71) Applicant (*for all designated States except US*): NEWBRIDGE NETWORKS CORPORATION [CA/CA]; 600 March Road, P.O. Box 13600, Kanata, Ontario K2K 2E6 (CA).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): HUANG, Chun-Chong [-/CA]; 43 Allenby Road, Kanata, Ontario K2K 2J9 (CA).

(74) Agent: MITCHELL, Richard, J.; Marks & Clerk, Station B, P.O. Box 957, Ottawa, Ontario K1P 5S7 (CA).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: CONNECTION ADMISSION CONTROL SYSTEM (CAC) FOR ATM NETWORKS

(57) Abstract

A method of carrying out connection admission control in an asynchronous network, comprises determining the virtual bandwidth in the set-up phase using a closed-form virtual bandwidth formula generalized from the exponential bounds for the queue length of incoming cells at a queuing point using a Martingale function, and permitting a connection to be established or maintained when the following condition is met: $\sum_j$ virtual bandwidth$_j$ x number of connections of type $j \leq C$($\equiv$ available bandwidth). The invention is useful for effecting connection admission control in ATM or like asynchronous data networks.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SG	Singapore
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LR	Liberia	SZ	Swaziland
CS	Czechoslovakia	LT	Lithuania	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	LV	Latvia	TJ	Tajikistan
DK	Denmark	MC	Monaco	TT	Trinidad and Tobago
EE	Estonia	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	UG	Uganda
FI	Finland	ML	Mali	US	United States of America
FR	France	MN	Mongolia	UZ	Uzbekistan
GA	Gabon	MR	Mauritania	VN	Viet Nam

CONNECTION ADMISSION CONTROL SYSTEM (CAC) FOR ATM NETWORKS

This invention relates to a asynchronous networks, such as asynchronous transfer mode (ATM) networks, and more particularly to a connection admission control system (CAC) for such networks.

Connection Admission Control is set of administrative procedures that must be taken to determine whether a request to establish a new ATM connection will be accepted or rejected.

Research on Connection Admission Control (CAC) for ATM networks has been intensively pursued ever since the ATM concept was introduced in the mid-1980s. As the ATM traffic management standards mature, the most promising approach is based upon estimating the amount of bandwidth (cell/sec) required, called virtual bandwidth, for supporting a connection request at the desired Quality of Service (QoS). There are also other less appealing approaches, such as the approach based upon the buffer congestion state at the moment of the connection request. These simplistic approaches lack the ability to take into account and to guarantee the standardized QoS (Quality of Service) definitions such as cell loss ratio, cell transfer delay and cell delay variation.

The available methods for calculating the virtual bandwidth are based on solving Markovian queueing systems. Some methods model each traffic source individually as Markovian Modulated Poisson Process (MMPP), Markovian Modulated Bernoulli Process (MMBP), Markovian Modulated Deterministic Process (MMDP), or Markovian Modulated Fluid-flow Process (MMFP). Other methods model the aggregate traffic as a whole as Markov Modulated Deterministic Process (MMDP), Gaussian process, or discrete-time batch Markovian arrival processes (D-BMAP). All these methods have the following disadvantages:

1. The methods are not real-time applicable. The requirement that the CAC (Connection Admission Controller) make the decision within a very short connection setup time does not allow for solving a complicated queueing problem.

2. The virtual bandwidth calculated for a connection depends on the number of other connections multiplexed in the queue as well as their traffic characteristics. As the number of connections in the queue and their traffic characteristics change all the time, these methods become infeasible for real-time application because the virtual bandwidth of all the existing connections has to be re-calculated all the time.

3. The approximation is somewhat heuristic and the accuracy varies quite significantly with the traffic characteristics of the connection.

4. The traffic characteristics specification and the QoS measures used in all the above methods are not fully aligned with those standardized in the ATM-Forum and ITU.

5. The above methods apply only to a single-buffer queue.

The second disadvantage has been addressed lately by a number of research groups discovering the independence of the virtual bandwidth on the number of connections in progress under certain system assumptions and performance criterion. In other words, the CAC unit simply checks the condition (1), namely

$$\sum_j \text{virtual bandwidth}_j \times \text{number of connections of type j} \leq C(\equiv \text{available bandwidth})$$

and the virtual bandwidth only has to be calculated once at the connection setup phase. In J. Y. Hui, "Resource allocation for broadband networks," IEEE J. Select. Areas Commun., vol.9, no. 7,pp.1598-1608, Sept. 1991, under bufferless system assumption (for studying burst level traffic management), the acceptance region is found to be approximately linear as stated in condition 1 under the performance criterion $P[\text{required bandwidth} > C] \approx \text{Chernoff bound} \leq \text{QoS}$. In R. J. Gibbens and P. J. Hunt, "Effective bandwidths for the multi-type UAS channel," Queueing Systems, 9:17-28, Sept. 1991., condition (1) is discovered in a single infinite-buffer system with MMFP sources under the performance criterion $P[X > B] \leq e^{-B\delta}$ for $B\delta >> 1$. where X is the number of cells in the buffer found by a typical arriving cell. Under the same performance criterion, condition (1) is also established for GI/G/1 queue in F. P. Kelly, "Effective Bandwidths at multi-class queues," Queueing Systems, 9:5-16, Sept. 1991. In

G. Kesidis, "Effective bandwidths for multiclass Markov fluids and other ATM sources," IEEE/ACM Tran. Networking, vol.1, no. 4, pp.424-428, Aug. 1993, the validity of condition (1) is further extended to more general sources under the single infinite-buffer system and the same performance criterion as above. However, no real-time application oriented method for calculating the virtual bandwidth in the context of condition (1) has been presented.

An object of the present invention is to alleviate the afore-mentioned problems.

According to the present invention there is provided a method of carrying out connection admission control in an asynchronous network, comprising the steps of determining the virtual bandwidth in the set-up phase using a closed-form virtual bandwidth formula generalized from the exponential bounds for the queue length of incoming cells at a queuing point using a Martingale function, and permitting a connection to be established or maintained when the following condition is met:

$$\sum_{j} \text{virtual bandwidth}_j \times \text{number of connections of type } j \leq C (\equiv \text{available bandwidth})$$

The incoming cells are normally monitored in the GI/G/1 queue (General Inter-arrival, General Server) in accordance with ATM terminology.

Martingale theory is described, for example, in J. F. C. Kingman, A Martingale inequality in the theory of queues, Proc. Camb. Phil. Soc. 59(1964) pp.359-361; and E. Buffet and N. G. Duffield, "Exponential upper bounds via Martingale for multiplexers with Markovian arrivals", Tech. Report DIAS-STP-92-16, School of Theoretical Physics, Dublin Institute for Advanced Studies, Dublin, Ireland.

The invention provides a virtual-bandwidth-based and fully standard-compliant real-time connection admission control system for ATM and like networks.

The connection admission decision is based on condition (1) described above, and the virtual bandwidth calculation is performed only once during the connection setup phase based on a closed-form formula. The present invention is therefore unlike the previous methods which use a heuristic approximation to obtain closed-form virtual

bandwidth formula. As a result, a very accurate estimation of the virtual bandwidth regardless of the traffic characteristics of the connection is obtained.

The CAC disclosed calculates the virtual bandwidth by taking into account the following standardized traffic parameters of a connection in the context of GCRA —Peak Cell Rate (PCR), Cell Delay Variation Tolerance (CDVT), Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS) as well as the following QoS measures—Cell Loss Ratio (CLR), Cell Delay Variation (CDV) and Cell Transfer Delay (CTD).

Finally, the CAC disclosed can be applied to multi-buffer queues if the arrival traffic satisfies a *general* condition.

This CAC is designed as a set of actions taken by the ATM network on a real-time basis at the call setup phase (or during call re-negotiation phase) in order to establish whether a Virtual Channel Connection (VCC) or a Virtual Path Connection (VPC) can be accepted or should be rejected. This CAC is in accordance with the traffic characteristics specification and the QoS measures defined in ATM Forum. However, this CAC can also be applied in any ATM network which is only partially compliant to the ATM Forum traffic management standard. Part of this CAC's functions can also be applied to estimate the capacity of any given ATM network.

The virtual bandwidth (VBW) calculation is the most important aspect of the CAC procedure as far as the bandwidth efficiency is concerned. The accuracy of virtual bandwidth directly affects the goodput of the network. To avoid under-estimating VBW, which may demerit the QoS guarantee, the tight upper bound of VBW is chosen to be the goal of the VBW algorithm.

The invention also provides a connection admission controller for an asynchronous network, comprising a control processor for controlling admission to the network, means for quantizing parameters descriptive of incoming connection requests, and a for memory storing pre-computed virtual bandwidths for each quantization level, said control processor determining the virtual bandwidth of a requested connection by

accessing the pre-computed value stored in said memory corresponding to the quantized parameter.

The invention will now be described in more detail, by way of example only, with reference to the accompanying drawings, in which:-

Figure 1 is a block diagram of an ATM network;

Figure 2 illustrates the basic connection admission procedure for an ATM network;

Figure 3 is a block diagram of a typical CAC controller;

Figure 4 illustrates the basic CAC procedure in accordance with the invention;

Figure 5 is a Traffic descriptor quantizer used in a table look-up method;

Fig. 6 shows a binary search algorithm used in implementing the present invention;

Fig. 7 shows the Queueing structure of Type 2 queueing point; and

Fig. 8 shows the similarity of shared-buffer system and dedicated-buffer system under balanced traffic.

In Figure 1, and ATM network 1 comprises a plurality of interconnected ATM switches 2, typically Newbridge Networks Corporation MainStreet™ 36170 ATM switches, having Connection Admission Control (CAC) units 3, which implement the CAC procedure to ensure that the desired Quality of Service (QOS) is maintained when a request to set up a new virtual channel connection (VCC), change the Service Category or traffic descriptor for an existing VCC, or remove an existing VCC previously admitted by CAC unit 3 is received.

The CAC unit 3 ensures that a VCC request is accepted only when sufficient resource is available to establish the connection at its required Quality of Service (QoS) and to maintain the agreed QoS of existing connections. Figures 2 and 4 show the procedure implemented by the CAC unit 3 in one embodiment, which will be described in more detail below.

When a connection request to the ATM network 1 at an incoming queueing point is received, unit 20 determines the parameters describing the connection, such as the Peak Cell Rate (PCR) and Sustained Cell Rate (SCR), and quantizes them into pre-defined levels. The sustained cell rate is defined in the ATM specifications as an upper bound on the conforming "average rate" of an ATM connection where the average rate is the number of cells transmitted divided by the duration of the connection. Unit 21 then looks up the virtual bandwidth from a look-up table, which stores pre-computed bandwidths for the quantized parameters. Unit 22 determines the amount of bandwidth available. Depending on the amount of bandwidth available, unit 23 decides whether to admit or reject the connection.

Figure 3 shows the CAC control processor 3 connected to look-up table 10 in the form of a RAM (Random Access Memory), which stores the pre-computed bandwidth values for the quantized parameters calculated using a Martingale function. The control processor 3 controls the operation of the ATM switch to admit or reject the requested connection.

In assessing each VCC request, the CAC unit 3 utilizes the following information in making the admission/refusal decision: the requested ATM-Forum-compliant QoS; values of the parameters in the VCC's ATM-Forum-compliant traffic descriptor, which is ATM Forum compliant; the requested UPC conformance definition; the routes (two routes for bidiretional point-to-point VCC and more routes for point-to-multipoint multicast VCC); and the user configurable booking factor.

The amount of network resource a VCC requires is represented in terms of a set of VBWs each corresponding to a congestion/queueing point on the routes. The VBW represents the minimum bandwidth a particular VCC requires at a queueing point to achieve the desired QoS.

The VBW is calculated using the VCC's traffic descriptor, the user configurable booking factor, the buffer structure of the queueing point, the buffer sizes and CLP_1 discard thresholds of the queueing point, and the target QoS. The VBW calculation also takes into account the statistical multiplexing gain across the VCCs sharing the same queueing point.

For CBR (Constant Bit Rate) traffic, the VBW calculation returns a value equal to the PCR (Peak Cell Rate) of the request. The VBW actually used by CAC is then equal to the PCR of the request divided by the CBR-service-category booking factor. Because of the CDV (Cell Delay Variation) introduced in the upstream nodes, the actual VBW may be higher than PCR.

For VBR traffic, the VBW of each VBR connection with traffic descriptor (PCR, SCR, MBS) is set to the value returned by the VBW algorithm divided by the VBR-service-category booking factor. The VBW algorithm will be described in more detail below.

VBW(Type of Queueing point, Traffic Descriptor, Buffer_size, CLR, C (necessary for Type 1 and 2 queueing points), Sfty_margin, resolution)

The Type of Queueing point can be type 1a, type 1b, type 2a, or type 2b (see Table 1 below). The Buffer_size is set to the depth of the corresponding priority buffer at the queueing point minus the corresponding CLP1 discard threshold. There are four priority buffers per switch port. The C is the total available bandwidth of the queueing point. The Sfty_margin represents a 10^{-sfty_margin} probability that the burst length is greater than MBS assuming geometrically distributed burst length. The recommended value for the Sfty_margin is 1. The resolution represents the convergence tolerance of the algorithm. The value returned by VBW (obtained by convergence of an iterative solution) is greater than the exact solution of the algorithm but the difference is no greater than resolution. The recommended value for the resolution is at most 1% of CLR.

Depending on the configuration, some queuing points receive both CBR and VBR traffic while others receive either CBR or VBR but not both. The VBW calculation for VBR VCC is different depending on whether the queuing point also receives the CBR traffic or not. The table below lists the association of the type of queuing point and the traffic it receives.

Table 1

VBW algorithm	Type of queueing point

VBW_{Type1}	Type 1 queueing point: Hub-card and UCS (single Diablo or Stealth) receive no CBR traffic
VBW_{Type2}	Type 2 queueing point DSC (multiple Diablos) receive no CBR traffic
VBW_{Type3}	Type 3 queueing point Hub-card, DSC or UCS receive CBR traffic also

A "diablo" referred to in the above table is a fast cell-switching chip, which forms part of the ATM switch. A "stealth" is a queueing device which receives cells admitted by the CAC unit for transmission to the network.

To alleviate the run-time processing burden, in an alternative embodiment, a Table Look-up method can be implemented. The VBWs of a limited number of representative traffic descriptors are pre-computed and stored in a table. The user specified traffic descriptor is then quantized into one of the predetermined traffic descriptor levels.

In the case of ABR traffic, the CAC unit 3 has to reserve at least MCR (Maximum Cell Rate) bandwidth for each ABR connection. This implies blocking of connection attempts when the smallest acceptable MCR of a connection request is greater than the available bandwidth at least one of the queuing points.

The safest CAC policy for ABR traffic is to adopt so called "Hard" allocation. Under "Hard" allocation, the MCR portion of the ABR connection is treated as CBR traffic. That is resources equivalent to MCR are reserved exclusively to the ABR connection and made unavailable to other connections, even when the ABR connection is idle. This policy is bandwidth inefficient.

The "Soft" allocation, on the other hand, permits the sum of the MCR values of all ABR connections to be greater than the actual allocated bandwidth for the ABR connections. When an ABR connection transmits at less than its MCR, the resources that it does not use are allocated using the network elements fairness policy among the other ABR connections. Monitoring and statistical engineering is needed to prevent excessive CLR. Assuming the average percentage of the inactive ABR connections and the average

burst size of each burst of a ABR connection, the actual required bandwidth for a given CLR and buffer size can be calculated as described below.

Figure 5 illustrates the Look-up method of determining virtual bandwidth. In Figure 5, the continuous spectrums of PCR and SCR of VBR service categories are divided into a number of levels with non-uniform step-size. Fig. 5 illustrates an example in which the SCR is divided. As demonstrated in Fig.5, each quantization level is represented by the maximum SCR of that level. When the PCR is divided, each PCR quantization level is also represented by the maximum PCR of that level. The user declared PCR and SCR are quantized according to properly designed quantization curve similar to that shown in Fig. 5.

The purpose of using table look-up method to obtain virtual bandwidth is to limit the number of traffic descriptor values to be considered by CAC unit 3 so that the RAM space required for CAC unit is under control, and the corresponding virtual bandwidth for each of these limited number of traffic descriptor values can be precomputed and stored in a table 10 (Figure 3), which would speed up run-time CAC processing.

The following system configuration information can be accessed by CAC at any time:

1) buffer sizes and CLP1-cell discard threshold values at all the queueing points,

2) total output link transmission capacity available for user connections at all the queueing points,

3) booking factors of all the service categories.

A table based on Figure 5 similar to table 2 shown below (referred to as the virtual bandwidth table) can then be built for each queueing point by using the virtual bandwidth described below to provide mapping between the discrete traffic descriptors (representatives of the quantization levels) and the corresponding virtual bandwidths. The virtual bandwidth table stays unchanged unless the system configuration is changed. If the system configuration changes, a new virtual bandwidth table at each queueing point to reflect such change must be rebuilt. However, no table updating is necessary during

normal operation. It will be understood by one skilled in the art that the "tables" referred to are in fact regions of RAM (Random Access Memory) reserved to store the listed data.

Table 2

quantization level	SCR	PCR	*VBW*
1	X	X	X
2	X	X	X
...			

In general, a virtual bandwidth table is maintained at each queueing point to store the precomputed virtual bandwidth values for all the (SCR,PCR) quantization levels. Suppose the same number of quantization levels and the same step-sizes are used at all the queueing points, the same buffer sizes and CLP1 discard thresholds are used at all the DSCs, and the same buffer sizes and CLP1 discard thresholds are used at all the Stealths. Then, one virtual bandwidth table with three VBW columns, one for multiplexing Diablo, one for DSC, and one for Stealth, can be shared by all the queueing points.

Another table (referred to as connection information table) is required by the Newbridge Networks Corporation MainStreet™ 36170 CAC procedure to process a VBR connection request has a format as shown in Table 3. Table 3 stores the information of all the connections currently through the switch. This table needs to be updated every time a new request is granted or an old connection is disconnected.

Table 3

connection	traffic descriptor	mux Diablo	switch Diablo	egress Stealth	associated quantization level of PCR and SCR at queueing points
1	(X,X,X)	1,1	3,3	3,4	(3, 5), (3, 6)
2	(X,X,X)	1,2	4,3	9,2	(7, 2), (7, 3)
...					

In the above table, the connection is the identifier of the connection by WGS CAC, the service category is the requested service category, the traffic descriptor represents the values of the traffic parameters (which traffic parameter to use depends on the service category), the mux Diablo is the mux Diablos (a Newbridge multiplexer) on the bidirectional connection path, the switch Diablo is the switch Diablos on the bidirectional connection path, the Stealth is the Stealths on the bidirectional connection path, and the associated quantization levels of PCR and SCR at queueing points are the mapped quantization levels of PCR and SCR at the traversed queueing points.

When CAC unit receives a request, it checks the traffic descriptor of the request against the Table 1 to find out the appropriate quantization level and retrieve the VBW at mux Diablo, DSC and Stealth. Following the CAC procedure, it can be determined if such request can be granted or rejected. If the request is granted, the change of the existing connection or the new connection will be written into Table 3.

When one of the connection recorded in Table 3 requests to be disconnected, its entry will be deleted from Table 3 and the virtual bandwidth retrieved from Table 2 according to the quantization level retrieved from Table 3 will be subtracted from VBW^{H}_{total}, VBW^{D}_{total}, and VBW^{U}_{total}.

There is a bandwidth overbooking for each connection because the smallest default traffic descriptor which is larger than the user specified one is used in the virtual bandwidth calculation. The larger the number of quantization levels, the smaller the overall bandwidth overbooking.

An example of a virtual bandwidth algorithm to calculate the virtual bandwidth of a connection request with traffic descriptor (*PCR* (cell/sec), *SCR* (cell/sec), *MBS* (cells)) will now be described.

The virtual bandwidth algorithms are modified algorithms of a basic algorithm to reflect the difference of the queueing structure as well as the receiving traffic pattern. The basic virtual bandwidth algorithm calculates the virtual bandwidth required by a

connection with a given traffic descriptor in a FIFO single server single buffer queue of a given buffer size B to achieve a specified cell loss ratio.

The traffic pattern assumption of a connection with traffic descriptor (PCR, SCR, MBS) is as follows. Assume the connection switches between active state and idle state alternately. During the active state, the connection generates cells periodically every $1/PCR$ seconds. Let Ma (cells) denote the average number of cells in a burst. The average inter-burst idle period length is given by

$$t_{idle} = \frac{M_a}{SCR} - \frac{M_a}{PCR} \text{ sec.}$$

Assume both the active and idle periods of each connection are geometrically distributed with mean M_a and $t_{idle} \cdot PCR$ (both in the unit of $\Delta = 1/PCR$ sec), respectively. However, a safety-margin factor η which relates Ma and MBS is introduced in such a way that the probability the burst length exceeds MBS is less than $10^{-\eta}$. The following relation can be derived:

$$Ma = \frac{1}{1-10^{-\eta/MBS}}$$

Let C (cell/sec) be the cell service rate of the queue. During $1/PCR$ second time (the cell inter-arrival time of the connection during active period), $s = C/PCR$ cells can be served. Let the total number of connections in the queue with the traffic descriptor (PCR, SCR, MBS) be L. Define $\sigma = s/L$. Assuming infinite buffer, the tail of the distribution of queue length q is upper bounded by the following explicit formula obtained via Martingales

$$\mathrm{P}[q \geq b] \leq \frac{a(1-(a+\sigma(1-a-d)))}{d(a+\sigma(1-a-d))}\left[\frac{(1-\sigma)(a+\sigma(1-a-d))}{\sigma(1-(a+\sigma(1-a-d)))}\right]^b \times$$
$$\left[\frac{1}{a+d}\left(\frac{d}{1-\sigma}\right)^{1-\sigma}\left(\frac{a}{\sigma}\right)^{\sigma}\right]^L$$

where $a = 1/(t_{idle} \cdot PCR)$ and $d = 1/M_a$.

Because $P[q \geq b]$ is always greater than the CLR under a finite buffer (of size b) system, the actual CLR is bounded below the right hand side of the above equation. Given the buffer size B and the desirable cell loss probability P, we can thus use $\sigma^* L \cdot PCR$ (cell/sec) safely as the aggregate virtual bandwidth, where σ^* satisfies the following transcendental function

$$p = \frac{a(1-(a+\sigma^*(1-a-d)))}{d(a+\sigma^*(1-a-d))}\left[\frac{(1-\sigma^*)(a+\sigma^*(1-a-d))}{\sigma^*(1-(a+\sigma^*(1-a-d)))}\right]^B \times \left[\frac{1}{a+d}\left(\frac{d}{1-\sigma^*}\right)^{1-\sigma^*}\left(\frac{a}{\sigma^*}\right)^{\sigma^*}\right]^L \qquad (1)$$

We increase L until $\sigma^* \cdot PCR \cdot L$ is less than the total available bandwidth C but for L+1 $\sigma^* \cdot PCR \cdot L$ is greater than C. The virtual bandwidth per connection— $\boxed{VBW_{basic}((PCR, SCR, MBS), C, B, CLR, \eta, \text{resolution})}$ is then $\boxed{\sigma^* \cdot PCR}$.

One can solve Eq.(1) iteratively. The virtual bandwidth is somewhere between SCR and PCR as shown in Fig. 6. As demonstrated in Fig. 6, we can solve σ^* by bisectional search with initial σ^* set as $\frac{\frac{1}{2}(\text{SCR}+\text{PCR})}{\text{PCR}} = \frac{1}{2}(1+\frac{a}{a+d})$. The search stops when the right hand side of the function is less than but within Δ (Δ represents the resolution of the virtual bandwidth) of the target CLR (i.e. 10^{-7}). The final $\sigma^* \cdot PCR$ is then the virtual bandwidth (with resolution Δ). Figure 6 shows a binary search procedure for solving this equation.

The VBR buffer of a Type 1 queueing point can be considered as a single buffer single server queueing system (CBR buffer is not in use). The virtual bandwidth of a VBR VC with traffic descriptor (SCR,PCR,MBS) is then

$$\boxed{VBW_{\text{Type1}} = VBW_{basic}(SCR, PCR, MBS, \text{Buffer\_size}, L_{\max}, CLR, \eta, \text{resolution})},$$

where $L_{\max}$ is the maximum L such that

$$L \times VBW_{basic}(SCR, PCR, MBS, \text{Buffer\_size}, L, CLR, \eta, \text{resolution}) \leq \text{egress bandwidth}$$

Fig. 7 shows the queueing structure of a Type 2 queueing point. Type 2 queueing point receives no CBR traffic and the output capacity C is totally available to the VBR VCs. As a result, the maximum VBR traffic that can be accepted and the corresponding statistical multiplexing gain can be predicted.

Theoretically, the existence of a unique virtual bandwidth for a traffic descriptor regardless of the number and the types of the connections in the system has only been proven for the single buffer single server queueing system. Such system is circuit-switching like, meaning that when admitting a connection, CAC only has to check

$$\sum_i N_i \cdot VBW_i \leq \text{available bandwidth} \quad ,(2)$$

where N_i is the number of connections of traffic descriptor i in the system and VBW_i is the virtual bandwidth of traffic descriptor i. For the queueing system of Fig. 8, there does not exist a unique virtual bandwidth for a traffic descriptor which is independent of the number and the types of the connections in the system. For instance, the required bandwidth of a connection is higher in the case where the arrival traffic concentrates in one of the buffers in Fig. 6 than in the case where the arrival traffic distributes evenly over all the buffers. In the latter case (balanced traffic case), the buffers are better utilized and the required bandwidth for a connection is reduced.

However, if the traffic of Fig.. 7 is more or less distributed evenly over all the buffers, the system becomes like a single buffer single server queueing system as demonstrated in Fig. 8 for a two buffer system, and Eq.(2) is approximately valid.

In the following, we assume that the arrival traffic to the system of Fig. 7 is distributed evenly over all the buffers. This is called the "balanced traffic assumption".

Suppose there exists a unique virtual bandwidth for this VC regardless of the number and the types of the VCs in the system of Fig. 7 under balanced traffic. By this assumption, the virtual bandwidth calculated under homogeneous traffic load (all the VCs

have the same traffic characteristics) must be the same as the virtual bandwidth calculated under heterogeneous traffic load (VCs have different traffic characteristics).

We now calculate the virtual bandwidth of a VC with a specified traffic descriptor (SCR,PCR,MBS) in a Type 2 queueing point. Assume that an equal number of such VCs are feeding each of the N buffers. Let L be the number of VCs per buffer. There are a total of $N \times L$ VCs.

If the statistical multiplexing gain across the VCs using different buffers is not taken into account, $VBW_{basic}(SCR, PCR, MBS, B, L, CLR, \eta, \text{resolution})$ is the required virtual bandwidth per connection. Denote $VBW_{basic}(SCR, PCR, MBS, B, L, CLR, \eta, \text{resolution})$ as $VBW_{\text{single buffer}}(L)$ for convenience.

There is a certain amount of bandwidth in $VBW_{\text{single buffer}}(L)$ that can be used by the VCs in the other buffers. Such amount of bandwidth is a fraction f of $VBW_{\text{single buffer}}(L) - SCR$. The amount of useful bandwidth to a VC from the $VBW_{\text{single buffer}}(L)$ of the VCs of the other buffers is then

$$\frac{(N-1) \times \dfrac{L \times (VBW_{\text{single buffer}}(L) - SCR) f}{(N-1)}}{L} = (VBW_{\text{single buffer}}(L) - SCR) f.$$

In the above equation, we assume that the useful bandwidth $f(VBW_{\text{single buffer}}(L) - SCR)$ of a VC is distributed uniformly over the VCs of the other buffers. The actual virtual bandwidth of a connection is

$$VBW_{\text{Type2}} = VBW_{\text{single buffer}}(L_{\max}) - f(VBW_{\text{single buffer}}(L_{\max}) - SCR),$$

where $L_{\max}$ is the maximum L such that

$$N \times L \times VBW_{\text{Type2}} \leq \text{Total available bandwidth of the congestion point}. \qquad (3)$$

The fraction f is a complicated function of the traffic descriptor. In the following, we present a simple approximating formula of f. Intuitively, the useful

bandwidth in the virtual bandwidth of a connection to the connections in other buffers decreases as traffic becomes smoother. Therefore, as the traffic burstiness $\frac{PCR}{SCR}$ decreases, f decreases, and vice versa. Actually, from the simulation results, the relation of f and $\frac{PCR}{SCR}$ appears to be

$$f = g(SCR) \cdot (1 - e^{-\beta \frac{PCR}{SCR}}).$$

The value of β, averaged over a large amount of simulation result, is 1.0666. We are unable to approximate $g(SCR)$ with a closed form expression, but can find a lower bound from the simulation results: $0.56 \cdot 1.03^{N} \le g(SCR)$. Substituting this lower bound for $g(SCR)$ in f, the actual virtual bandwidth of a connection becomes

$$\boxed{\begin{aligned} VBW_{\text{Type2}} &= VBW_{\text{single buffer}}(L_{\max}) \\ &\quad -(VBW_{\text{single buffer}}(L_{\max}) - SCR) \times 0.56 \times 1.03^{N} \times (1 - e^{-\beta \frac{PCR}{SCR}}) \end{aligned}} \qquad (4)$$

In the case where Eq.3 can not be satisfied when L=1, there are only three or less VCs with traffic descriptor (PCR, SCR, MBS) that can be accepted. In this case, the number of buffers that are actually in use is less than 4. The $L_{\max}$ in Eq.A4 now becomes the maximum L such that

$$\text{Number of buffer in use} \times L \times VBW_{\text{Type2}} \le \text{Total available bandwidth of the congestion point}$$

At a Type 3 queueing point, the available bandwidth for VBR traffic is time-variant depending on how much bandwidth the CBR traffic uses. Therefore, at a Type 3 queueing point, the maximum VBR traffic acceptable and its corresponding statistical multiplexing gain across the connections are also time-variant depending on how much bandwidth the CBR traffic books. To calculate the VBW at system setup without under-estimation, a method different from that used in the Type 1 and 2 queueing points must be developed.

Because the statistical multiplexing gain across the connections are unforeseen, it must be ignored in order to prevent under-estimation. Therefore, the VBW of a VBR connection with a specified traffic descriptor (SCR,PCR,MBS) is calculated as

$$\boxed{VBW_{\text{Type 3}} = VBW_{\text{basic}}(SCR, PCR, MBS, \text{Buffer\_size}, L = 1, CLR, \eta, \text{resolution})}.$$

The ABR virtual bandwidth algorithm is a modified version of the VBR virtual bandwidth algorithm. The PCR in the VBR VBW algorithm is now replaced by the MCR. The SCR in the VBR VBW algorithm is now replaced by the product of MCR and the average percentage of inactive ABR connections. The MBS in the VBR VBW algorithm is not used. Instead, the Ma (see Sec. A.1) in the VBR VBW algorithm is set to the average burst size.

The elementary functions required to compute the virtual bandwidth are as follows:

1) add ($A+B$)

2) subtract ($A-B$)

3) multiply ($A \times B$)

4) divide ($A \div B$)

5) power (A^B)

6) floor (the largest integer no greater than A)

The data structure of the real number represented by two integers is

```
typedef struct Real_Number {
    int        Mantissa;
    int        Exponent
} Real;
```

An R of type Real corresponds to the real number $R.Mantissa \times 10^{R.Exponent}$. In the following, the design of the above 5 elementary functions which operate on the data type Real will be described. For a 32 bit machine (e.g. SPARCstation), the maximum Mantissa and Exponent is $2^{31}-1=2147483647$. However, we limit the maximum of Mantissa to 9999. In other words, each number expressed in Real format has 4 significant decimal digits. The reason to limit the maximum Mantissa to only 9999 is to avoid overflow caused by multiplying two Real numbers with the maximum Mantissa. For example, $99999 \times 99999 = 9999800001 > 2^{31}-1$ but $9999 \times 9999 = 99980001 < 2^{31}-1$. The Mantissa of a Real number is always maintained within 1000 and 9999 to allow maximum accuracy.

To add two Real numbers A and B, the one (say A) with larger absolute value is adjusted (A.Mantissa multiplied by $10^{A.Exponent-B.Exponent}$) so that its Exponent is equal to that of B. Then the two Real numbers can readily be added. In this way, the accuracy will not be detrimented. To avoid overflow caused by multiplying $10^{A.Exponent-B.Exponent}$, the Real number B will be ignored if $A.Exponent - B.Exponent \geq 5$ unless A=0.

Similar to ADD, the one (say A) with larger absolute value is adjusted so that its Exponent is equal to that of B before the subtraction is executed. The Real number B will be ignored if $A.Exponent - B.Exponent \geq 5$ unless A = 0;

To multiply A and B, we simply calculate the resultant Mantissa as $A.Mantissa \times B.Mantissa$ and the resultant Exponent as $A.Exponent + B.Exponent$.

To divide A with B, the Real number of A is first adjusted so that its Mantissa is within 100000000 and 999999999 to obtain maximum accuracy. Then, we calculate the resultant Mantissa as $A.Mantissa(adjusted) \div B.Mantissa$ and resultant Exponent as $A.Exponent(adjusted) - B.Exponent$.

First of all, if B=0, $A^B = 0$, regardless of A; if A=0, $A^B = 0$, regardless of B; and if A=1, $A^B = 1$, regardless of B.

Since $A^B = e^{B \times \log_e A}$, we implement $\log_e R$ and e^R two elementary functions as the way to calculate A^B. Two power series approximations are used to implement $\log_e R$ and e^R:

$$\log_e(1+R) = \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j} \times R^j,$$

where $-1 < R \leq 1$;

$$e^R = \sum_{j=0}^{\infty} \frac{X^j}{j!}.$$

The above two functions can be easily implemented by using ADD, SUBTRACT, MULTIPLY and DIVIDE already defined. To utilize $\log_e(1+R)$ formula, A needs to be adjusted so that $A = M \times 10^E$, where M is within 0 and 2. Then

$$\log_e A = \log_e M + E \log_e 10 = \log_e(1 + (M-1)) + E \log_e 10,$$

which can readily be calculated. Note that $\log_e 10 = 2.3026$ is a constant.

To obtain the maximum accuracy of e^R when $R < -1$, we calculate $\frac{1}{e^{-R}}$ instead because e^{-R} is more likely to be dominated by the first few terms in the power series expansion.

The described invention can be implemented on a general purpose computer, for example, a SPARCstation, although with the aid of the look-up table described with reference to Figure 5, the CAC unit 3 can be implemented using microprocessor technology.

The present invention thus provides an improved method of controlling the access to an asynchronous data network, for example, an asynchronous transfer mode network, following a request to set up a new virtual channel connection (VCC), change the Service Category or traffic descriptor for an existing VCC, or remove an existing VCC.

Claims:

1. A method of carrying out connection admission control in an asynchronous network, comprising the steps of determining the virtual bandwidth and permitting a connection to be established or maintained when the following condition is met:

$$\sum_j \text{virtual bandwidth}_j \times \text{number of connections of type j} \leq C(\equiv \text{available bandwidth})$$

characterized in that the virtual bandwidth is determined in the set-up phase using a closed-form virtual bandwidth formula generalized from the exponential bounds for the queue length of incoming cells at a queuing point using a Martingale function.

2. A method as claimed in claim 1, characterized in that a look-up table is stored in a memory for each queueing point, said look-up table storing a limited number of traffic descriptors and computed virtual bandwidths therefor, and the virtual bandwidth during connection admission control is determined by retrieving said precomputed values therefrom.

3. A method as claimed in claim 2, characterized in that a common table is used for all queuing points.

4. A method as claimed in claim 2, characterized in that the values of the traffic descriptors are stored as a plurality of quantization levels.

5. A method as claimed in claim 4, characterized in that said quantization levels have a non-uniform step.

6. A method as claimed in claim 4, characterized in that said traffic descriptors are the sustained cell rate (SCR) and peak cell rate (PCR).

7. A method as claimed in claim 1, characterized in that the sustained cell rate (SCR) and peak cell (PCR) rate at a queueing point are measured, and the virtual bandwidth is calculated from the SCR and PCR using said Martingale function.

8. A method as claimed in claim 7, characterized in that the virtual bandwidth is calculated as $\boxed{\sigma^* \cdot PCR}$ where $\sigma = s/L$, $s = C/PCR$, s = number of cells/sec, L is the total

number of connections in the queue with the traffic descriptor under consideration, and σ^* satisfies the equation)

$$p = \frac{a(1-(a+\sigma^*(1-a-d)))}{d(a+\sigma^*(1-a-d))}\left[\frac{(1-\sigma^*)(a+\sigma^*(1-a-d))}{\sigma^*(1-(a+\sigma^*(1-a-d)))}\right]^B \times$$
$$\left[\frac{1}{a+d}\left(\frac{d}{1-\sigma^*}\right)^{1-\sigma^*}\left(\frac{a}{\sigma^*}\right)^{\sigma^*}\right]^L$$

where $a = 1/(t_{idle} \cdot PCR)$ and $d = 1/M_a$, and t_{idle}, M_a.

9. A connecting admission control unit comprising means for determining the virtual bandwidth and permitting a connection to be established or maintained when the following condition is met:

$$\sum_j \text{virtual bandwidth}_j \times \text{number of connections of type } j \leq C(\equiv \text{available bandwidth})$$

characterized in that means are provided for determining the virtual bandwidth in the set-up phase using a closed-form virtual bandwidth formula generalized from the exponential bounds for the queue length of incoming cells a queuing point using a Martingale function.

10. A connecting admission control unit as claimed in claim 9, characterized in that it comprises a memory for storing a look-up table of pre-computed virtual bandwidths for a limited number of traffic descriptors.

11. A connecting admission control unit as claimed in claim 9, characterized in that said memory stores said traffic descriptors as predetermined quantization levels.

12. A connection admission controller for an asynchronous network, comprising a control processor for controlling admission to the network, characterized in that means are provided for quantizing parameters descriptive of incoming connection requests, a memory stores pre-computed virtual bandwidths for each quantization level, and said control processor determines the virtual bandwidth of a requested connection by accessing the pre-computed value stored in said memory corresponding to the quantized parameter.

13. A connection admission controller as claimed in claim 12, characterized in that said pre-computed values are calculated using a closed-form virtual bandwidth formula generalized from the exponential bounds for the queue length of incoming cells a queuing point using a Martingale function.

14. A connection admission controller as claimed in claim 13, characterized in that said quantization levels are arranged in non-uniform steps.

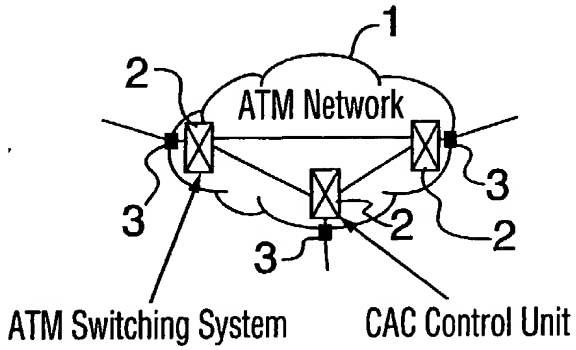
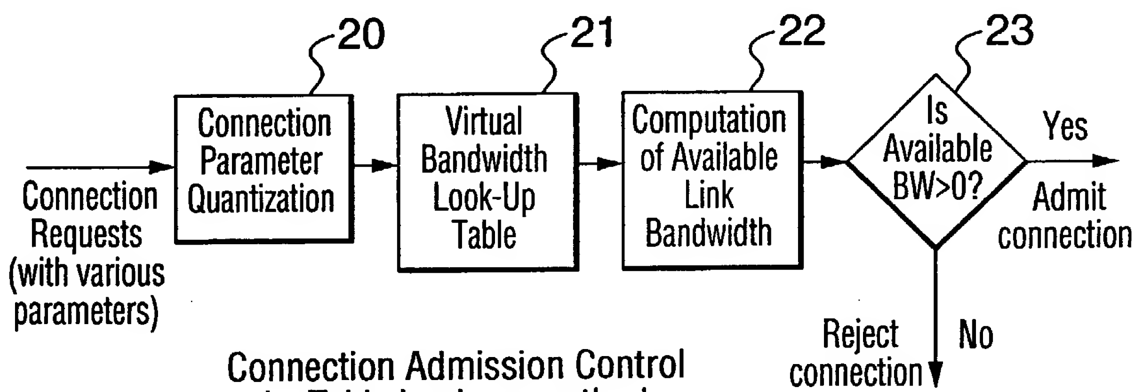


FIG. 1



Connection Admission Control by Table Look-up method

FIG. 2

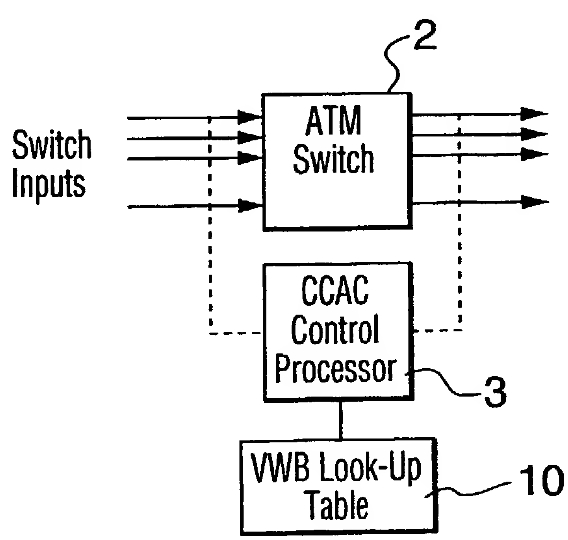


FIG. 3

A typical implementation of CAC controller

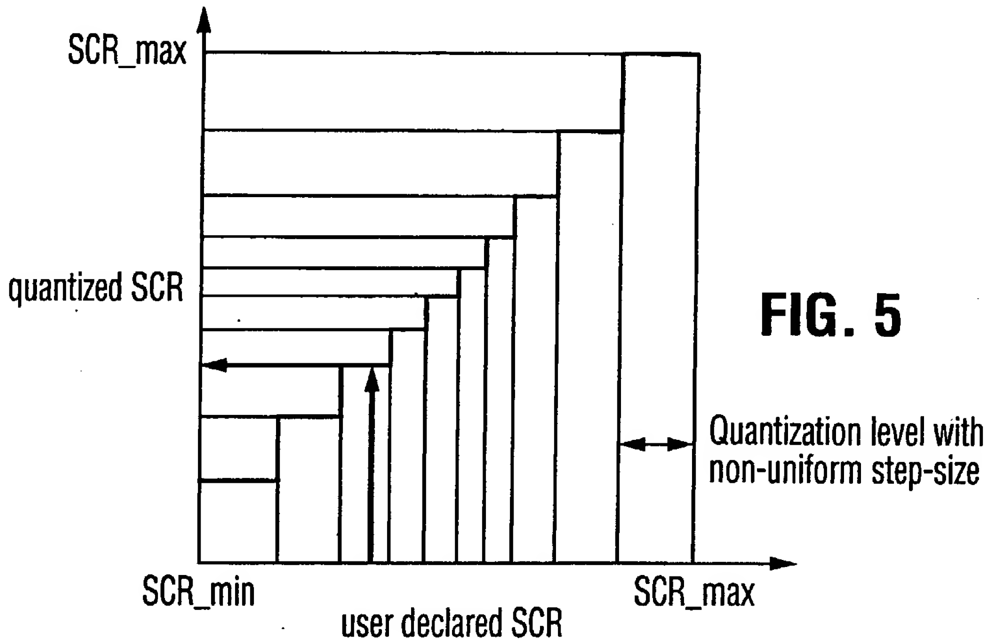

SCR_max
quantized SCR
Quantization level with non-uniform step-size
SCR_min
user declared SCR
SCR_max

FIG. 5

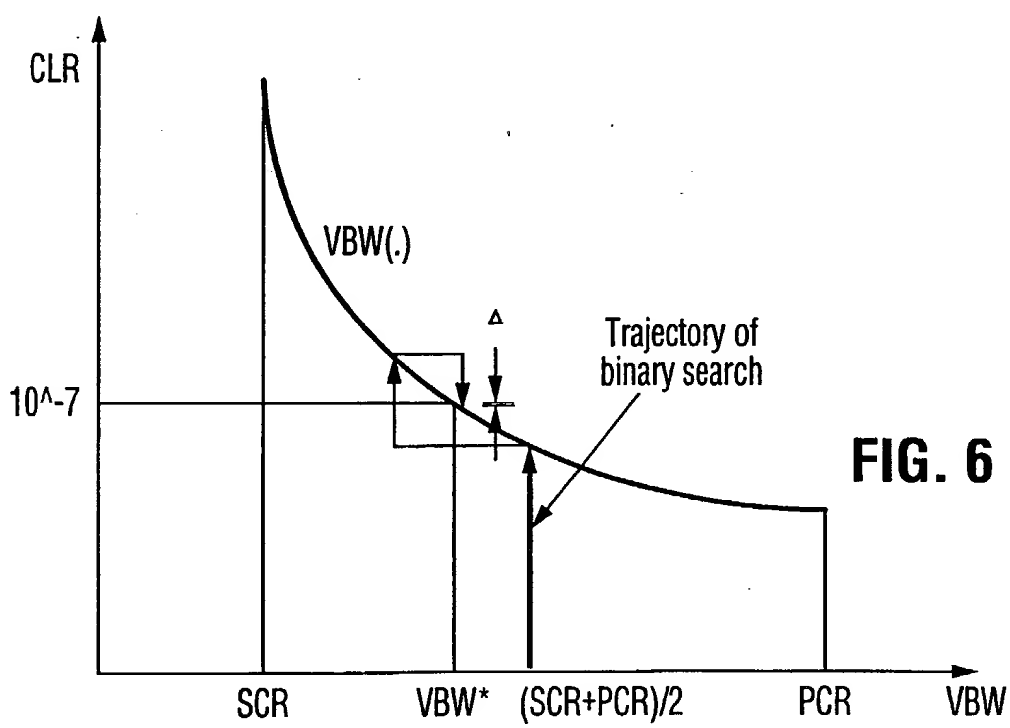

CLR
VBW(.)
Δ
Trajectory of binary search
10^-7
SCR
VBW*
(SCR+PCR)/2
PCR
VBW

FIG. 6

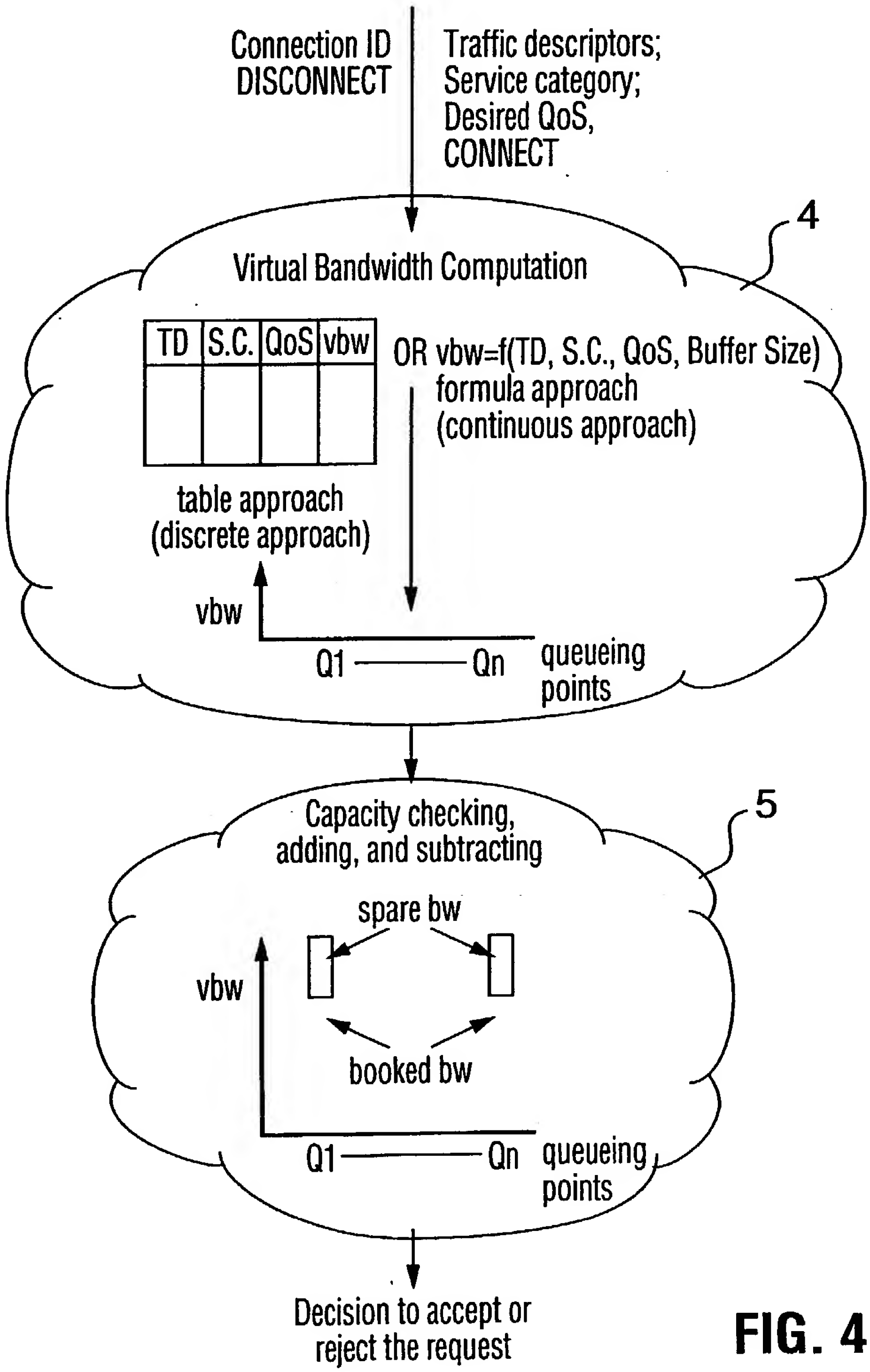

Connection ID
DISCONNECT
Traffic descriptors;
Service category;
Desired QoS,
CONNECT
4
Virtual Bandwidth Computation
TD
S.C.
QoS
vbw
OR vbw=f(TD, S.C., QoS, Buffer Size)
formula approach
(continuous approach)
table approach
(discrete approach)
vbw
Q1
Qn
queueing
points
Capacity checking,
adding, and subtracting
5
spare bw
vbw
booked bw
Q1
Qn
queueing
points
Decision to accept or
reject the request

FIG. 4

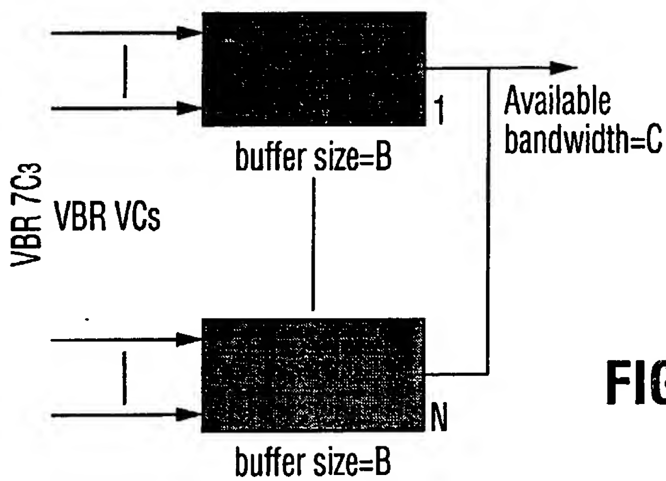

1
Available
bandwidth=C
buffer size=B
VBR 7C3
VBR VCs
N
buffer size=B

FIG. 7

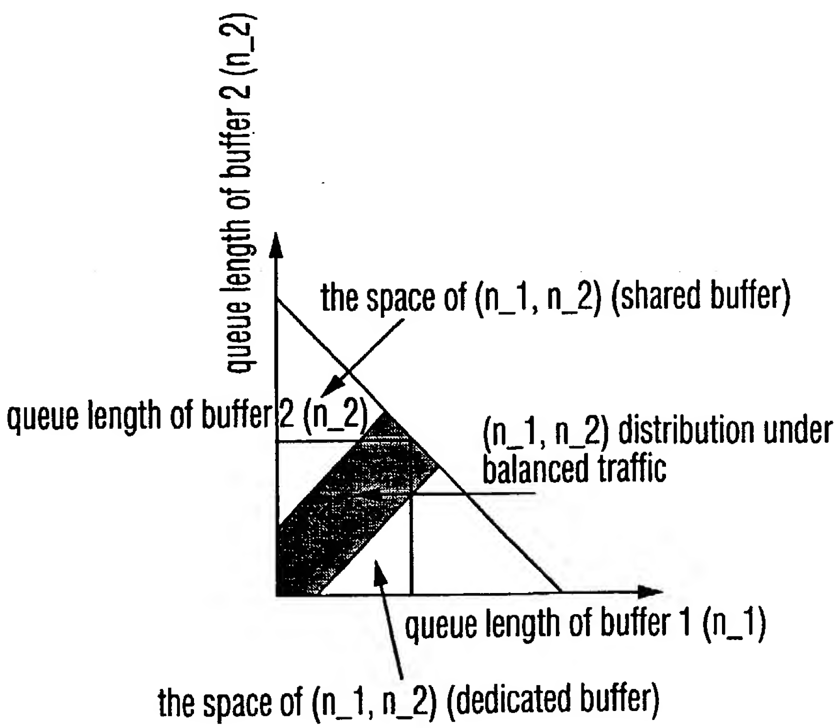

queue length of buffer 2 (n_2)
the space of (n_1, n_2) (shared buffer)
queue length of buffer 2 (n_2)
(n_1, n_2) distribution under
balanced traffic
queue length of buffer 1 (n_1)
the space of (n_1, n_2) (dedicated buffer)

FIG. 8